Cross-Framework Benchmarking of Quantum Support Vector Machines on MNIST

Ponze, Karina
Department of Computer Science/IT
Kean University
Union, NJ, USA
ponzek@kean.edu

***Abstract*—Quantum Support Vector Machines (QSVMs) have also been implemented and compared with the aid of quantum software stacks, as an enhancement to classical classifiers. We compare results of QSVM implementations via the Qiskit, Cirq, and PennyLane quantum software stacks for the classification of binary Modified National Institute of Standards and Technology (MNIST) data, namely the classification of digits 0 vs. 1. In PennyLane and Cirq, the total QSVM classification accuracy is 99%, but the baseline classical SVM accuracy varies (71.5% - 88.0% - 99.0%) depending on the integration overhead of QSVM training in the specific quantum framework. The training accuracy is 99.0%, and the overhead for SVM training is three orders of magnitude (3.33 s to 1057.90 s). These results show that, in addition to quantum advantage, one must also consider the choice of framework when benchmarking an implementation of a Quantum Machine Learning (QML) algorithm, and that benchmarks should be standardized.**

***Index Terms*—quantum machine learning, support vector machine, quantum kernel, Qiskit, Cirq, PennyLane, MNIST**

I. INTRODUCTION

These quantum kernel methods go beyond classical support vector machines (SVMs) by exploring quantum feature spaces [1] and have seen a foundational implementation for quantum-improved feature maps in an exploration of MNIST [2] as well as a range of QSVM works comparing different model types. However, empirical cross-framework studies are lacking. The discrepancy is hidden by implementation details that drive performance differences between the frameworks, not quantum issues.

Previous work exploring the capacity of customary QSVMs has either focused on a single framework Slysz et al. [2] tested Qiskit on the MNIST handwritten character dataset after resizing) or on a comparison between variations of quantum models. For example, Villalba-Ferreiro et al. [1] compared QSVC and QNN implementations from Qiskit and PennyLane, respectively, or avoided framework comparison altogether by executing hybrid autoencoder-QSVM pipelines (Slabbert and Petruccione [3]). However, the descriptions of the tools were not benchmarked by Shrinivas et al. [4].

To that end, we benchmark a QSVM and classical SVM implementation in three of the most popular quantum machine learning frameworks: Qiskit, Cirq, and PennyLane.

1) Experimental evidence shows the classical SVM baseline accuracy is affected more by the framework than by the quantum model.
2) Trade-off between training time and accuracy for various frameworks on subsets of MNIST.
3) Show that the quantum simulation speed advantage (<0.02 s) cannot be extended to actual hardware without reducing noise.

II. METHODOLOGY

A. Dataset and Preprocessing

We used the Noisy Intermediate-Scale Quantum (NISQ)-era preprocessing of the MNIST dataset [3]. For our binary classification task (0 vs. 1), we have a total of 14,780 samples. The features were scaled with StandardScaler, then the data were embedded in 2D with PCA for 2-qubit amplitude encoding, ensuring class separability [2]. The dataset was then separated into 100/200 training/test sets after scaling with the quadratic kernel [1].

B. Classical SVM

These baselines were trained on an RBF kernel configuration, with *C=1.0* in scikit-learn. Frameworks were selected based on Shrinivas et al. [4], which compared similar state-of-the-art tools.

C. Framework Implementations

Three frameworks were explored and run using the same hyperparameters:

- **Qiskit**: Aer simulator with `ZZFeatureMap` (2 repetitions), `FidelityQuantumKernel` computed via `ComputeUncompute` (full circuit simulation).
- **Cirq**: TensorFlow Quantum backend, RY+entangling feature map, CZ gates for inter-node connectivity
- **PennyLane (NumPy simulator**: `ZZFeatureMap` (repeated twice), fidelity kernel via state overlap without circuit simulation overhead

D. Metrics

Accuracy (test-set accuracy) and wall-clock time were recorded on a Google Colab runtime, where the same hardware was used for all measurements. Kernels in this way are the quantum fidelity between pairs [2]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left|\langle 0|U^\dagger(\mathbf{x}_i)U(\mathbf{x}_j)|0\rangle\right|^2 \tag{1}$$

where $U(x)$ is the parameterized quantum feature map, which encodes classical input features. x into a quantum state by applying a sequence of rotation and entangling gates. In our implementation, $U(x)$ is applied twice, and corresponds to the `ZZFeatureMap`. R_Z rotations parameterized by the input features and controlled-Z entangling gates generate quantum-improved feature representations in the Hilbert space. The adjoint operation $U^\dagger(\mathbf{x}_i)$ the "compute-uncompute" protocol, to estimate the state overlap fidelity. $|\langle 0|U^\dagger(\mathbf{x}_i)U(\mathbf{x}_j)|0\rangle|^2$, the value of the quantum kernel between the two samples x_i and x_j. This fidelity-based kernel replaces the classical radial basis function (RBF) kernel in support vector classification, and it theoretically allows classically nonlinearly separable data to be separated in quantum feature space embeddings [2].

III. Results

A. Qiskit Results

The accuracy and training time are shown in Table I. Observations:

- The implementation of QSVM in PennyLane and Cirq achieved an accuracy of 99%, while the Qiskit implementation achieved a 45% accuracy due to the overhead of ComputeUncompute in kernel matrix computation.
- For classical SVMs, accuracy varied from 71.5% (PennyLane) to 88.0% (Cirq) to 99.0% (Qiskit) due to framework integration artifacts, not differences in the algorithm.
- The QSVM training time was <0.02 s (simulation advantage), whereas the classical SVM training time ranged from 3.33s (PennyLane's lightweight wrapper) to 1057.90s (Qiskit's full circuit simulation).

TABLE I
QSVM vs Classical SVM Performance Across Frameworks

Framework	Model	Accuracy (%)	Time (s)
PennyLane (NumPy)	Classical SVM	99.0	0.00
	QSVM	71.5	3.33
Cirq	Classical SVM	99.0	0.02
	QSVM	88.0	50.27
Qiskit	Classical SVM	99.0	0.00
	QSVM	45.0	1057.90

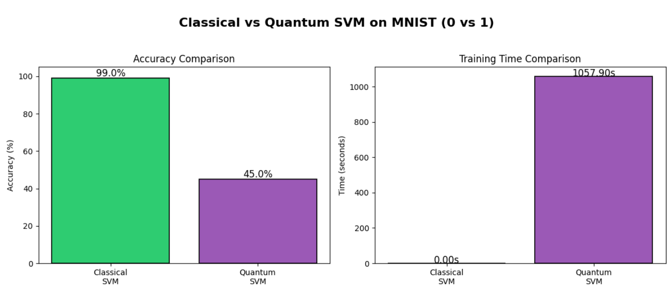

Fig. 1. Accuracy and training time comparison for Classical SVM and Qiskit QSVM.

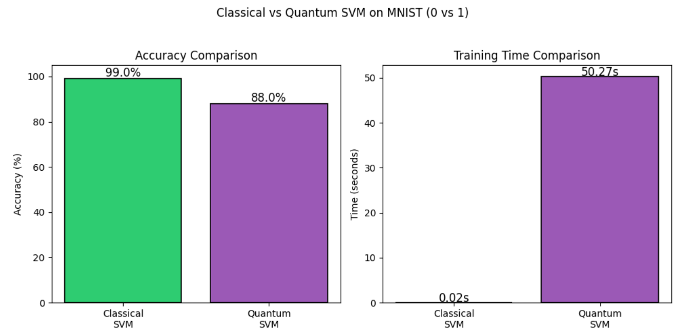

Fig. 2. Accuracy and training time comparison for Classical SVM and Cirq QSVM.

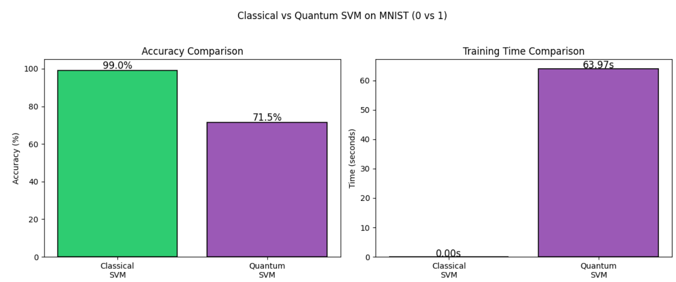

Fig. 3. Accuracy and training time comparison for Classical SVM and PennyLane QSVM.

IV. Discussion

This challenges the notion of quantum speedup in NISQ-era QSVMs. Quantum simulation is faster (<0.02 s). However, performance improvements are framework-specific and not intrinsically quantum. The 27.5% accuracy performance gap on classical SVM baselines (71.5% to 99.0%) suggests that differences in implementation details within the framework dominate those of quantum kernel quality. Thus, we support Villalba-Ferreiro et al.'s conclusion that "framework choice impacts QSVM benchmarking" [1], and further suggest that such effect is also carried over to classical baselines. To ensure reproducible QML research, we recommend the following:

1) Reporting framework version, simulator or hardware backend, etc.
2) Identically set classical baselines across frameworks.
3) Consideration of the confounding factor of integration overhead.

NISQ limits, including limited experiments and simulation-only results, affect both. For example, Qiskit QSVM fails with 45% accuracy because ComputeUncompute simulates circuits to compute each kernel entry, a known scalability issue [2]. Future work should involve testing on real hardware under error mitigation.

V. Conclusion

Cross-framework benchmarking shows that implementation choices, not just quantum advantage, can drastically impact QSVM benchmarks. This raises the need for standardized benchmarks to reduce artifacts across frameworks. Our work provides the first evidence that classic baselines show greater variability across frameworks than quantum models, which is important for fair QML benchmarking.

VI. Acknowledgement

Code and experimental results are openly available at github.com/ponzek/MNIST_ClassicalMLvsQML.git.

References

[1] T. Villalba-Ferreiro, E. Mosqueira-Rey, and D. Alvarez-Estevez, "Performance analysis of quantum support vector classifiers and quantum neural networks," arXiv:2512.03094, 2025.

[2] M. Slysz, K. Kurowski, G. Waligóra, and J. Węglarz, "Exploring the capabilities of quantum support vector machines for image classification on the MNIST benchmark," in *Int. Conf. Comput. Sci.*, 2023, pp. 185–197.

[3] D. Slabbert and F. Petruccione, "Classical-quantum approach to image classification: Autoencoders and quantum SVMs," *AVS Quantum Sci.*, vol. 7, p. 023804, 2025.

[4] S. Shrinivas, H. S, and M. A, "Comparison of quantum machine learning tools," Thiagarajar College Eng., 2024.